



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/976,726	10/12/2001	Jason King	5150-58700	2464
35690	7590	01/06/2006	EXAMINER	
MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.			ZHOU, TING	
P.O. BOX 398			ART UNIT	
AUSTIN, TX 78767-0398			PAPER NUMBER	

2173

DATE MAILED: 01/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>		<b>Applicant(s)</b>	
	09/976,726		KING ET AL.	
	<b>Examiner</b>		<b>Art Unit</b>	
	Ting Zhou		2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 04 November 2005.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-25,32-44,46-60 and 62-89 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-25,32-44,46-60 and 62-89 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)             | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)    | Paper No(s)/Mail Date. _____  |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____   | 6) <input type="checkbox"/> Other: _____                                    |

### **DETAILED ACTION**

1. The amendment filed on 4 November 2005 have been received and entered. The applicant has cancelled claims 26-30 and added new claims 72-89. Claims 1-25, 32-44, 46-60 and 62-89 as amended are pending in the application.
2. Claims 16-18 and 71 were previously indicated as allowable subject matter, which were objected to as being dependent upon a rejected base claim in the office action dated 25 July 2005.

### ***Allowable Subject Matter***

3. The indicated allowability of claims 16-18 and 71 are withdrawn in view of the newly discovered reference(s) to Cain et al. U.S. Patent 5,651,108. Rejections based on the newly cited reference(s) follow.

### ***Claim Rejections - 35 USC § 112***

The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 16, 71, 80-81 and 87 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the

Art Unit: 2173

relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The limitations “the first node is *not operable* to receive the first user interface event” in claims 16 and 71 and “the first portion of graphical code *does not execute* in response to the first programmatic event being generated”, in claim 87, are not positively recited in the specification of the present application. Although the specification states that after dynamically registering an event during execution of the graphical program, the graphical program is operable to receive the first user interface event, the specification does not explicitly recite that the exclusion of the first node from receiving the first user interface event before dynamic registration. “Any negative limitation or exclusionary proviso must have basis in the original disclosure. If alternative elements are positively recited in the specification, they may be explicitly excluded in the claims....The mere absence of a positive recitation is not basis for an exclusion” (MPEP 2173.05 (i)). Therefore, there is no positively recited basis for the negative limitations of claims 80 and 81.

5. Claims 80 and 81 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. The limitations “wherein said receiving the third user input explicitly specifying the one or more user interface events to configure for the first node *does not* include receiving user input specifying a connection between the first node and a second node” and “wherein said receiving the third user input explicitly specifying the one or more user interface events to

configure for the first node is performed *independently* of configuring other nodes in the block diagram of the graphical program”, of claims 80 and 81 respectively, are not positively recited in the specification of the present application. “Any negative limitation or exclusionary proviso must have basis in the original disclosure. If alternative elements are positively recited in the specification, they may be explicitly excluded in the claims....The mere absence of a positive recitation is not basis for an exclusion” (MPEP 2173.05 (i)). The specification does not explicitly recite the exclusion of receiving user input specifying a connection between the first node and a second node and configuring other nodes in the block diagram of the graphical program. Therefore, there is no positively recited basis for the negative limitations of claims 80 and 81.

### ***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-7, 10-25, 32-44, 46-60, 62-79 and 82-89 are rejected under 35 U.S.C. 102(b) as being anticipated by Cain et al. U.S. Patent 5,651,108 (hereinafter “Cain”).

Referring to claims 1, 19 and 32, Cain teaches a method and memory medium comprising creating a graphical user interface for the graphical program in response to first user input (for example, users opening the graphical user interface shown in Figure 4A for providing event-

based, interactive visual-programming) (column 3, lines 18-45 and column 7, lines 18-32); displaying a first node for receiving user interface events in a block diagram for the graphical program in response to second user input (displaying a button in response to user selection for receiving methods which will execute in response to programmed user events) (column 3, lines 18-65, column 7, line 63-column 8, line 14 and column 10, line 50-column 12, line 46); receiving third user input explicitly specifying one or more user interface events to configure for the first node (users can change or attach new user interface events which the button will respond to via input on the graphical user interface, as described with relation to Figures 4D-4E) (column 11, line 30-column 12, line 46); configuring the first node to receive the one or more user interface events explicitly specified by the third user input during execution of the graphical program (during execution of the program, i.e. in the run mode, the button object will respond to user defined interface events such as generation of a push-button event upon clicking on the button) (column 11, line 30-column 12, line 46); and associating one or more portions of graphical code with the first node in response to fourth user input, wherein each portion of graphical code comprises one or more nodes for responding to one or more of the user interface events which the first node is configured to receive (attaching program code to a button such that the code executes when the button is clicked) (column 10, line 48-column 11, line 3).

Referring to claims 23 and 86, Cain teaches a method comprising creating a graphical user interface for the graphical program in response to user input (for example, users opening the graphical user interface shown in Figure 4A for providing event-based, interactive visual-programming) (column 3, lines 18-45 and column 7, lines 18-32); creating a block diagram for the graphical program in response to user input, wherein creating the block diagram comprises

creating a first portion of graphical code in response to user input, wherein the first portion of graphical code includes one or more nodes for responding to user interface events (creating a block diagram such as the diagram shown in Figure 5F for example, via a plurality of selected objects such as buttons for receiving methods which will execute in response to programmed user events) (column 3, lines 18-65, column 7, line 63-column 8, line 14 and column 10, line 50-column 12, line 46); receiving user input explicitly specifying one or more user interface events to associate with the first portion of graphical code (attaching program code specifying a response to user interface events such as users clicking the button is attached to the button) (column 3, lines 18-65 and page 10, line 50-column 11, line 29); and configuring the first portion of graphical code to execute in response to the one or more explicitly specified user interface events being generated during execution of the graphical program (upon the specified user interface event being generate, i.e. user clicking on the button, the program code attached to the button is executed to perform a function such as displaying a “Hello. World!” message in a dialog box) (column 3, lines 18-65 and page 10, line 50-column 11, line 29).

Referring to claims 36, 53 and 66, Cain teaches a method, memory medium and system comprising memory for storing program instructions (Figure 1A); a processor coupled to the memory (Figure 1A); and a display device (Figure 1A); wherein the processor is operable to execute program instructions stored in the memory to: display a first node in a block diagram of the graphical program in response to user input (displaying a button in response to user selection for receiving methods which will execute in response to programmed user events) (column 3, lines 18-65, column 7, line 63-column 8, line 14 and column 10, line 50-column 12, line 46); associate a first portion of graphical source code with the first node in response to user input

(attaching program code to buttons) (column 10, lines 50-58), wherein the first portion of graphical source code associated with the first node comprises a plurality of interconnected nodes (nodes can be interconnected with, i.e. contained within other nodes, as shown by the data diagram of interconnected nodes shown in Figure 5G) (column 14, line 56-column 17, line 24); associate a first user interface event with the first node in response to user input explicitly specifying the first user interface event (users can attach or associate user interface events which the button will respond to via input on the graphical user interface, as described with relation to Figures 4D-4E) (column 11, line 30-column 12, line 46); and configure the first portion of graphical source code associated with the first node to execute in response to the first user interface event associated with the first node being generated during execution of the graphical program (program code can be attached to buttons so that the code executes in response to specified user interface events such as users clicking the button) (column 3, lines 18-65 and column 10, line 50-column 12, line 46).

Referring to claims 2, 20 and 33, Cain teaches wherein the first node comprises one or more sub-diagrams (objects contained within other objects), wherein associating the one or more portions of graphical code with the first node comprises displaying each portion of graphical code within one of the sub-diagrams of the first node (objects contained with other objects inherits the events from the parent object) (column 14, line 56-column 17, line 24).

Referring to claims 3, 21, 34, 72-73, Cain teaches wherein the receiving the user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input explicitly specifying one or more user interface events to which each of the sub-diagrams of the first node corresponds; wherein for each portion of graphical



code, displaying the portion of graphical code within one of the sub-diagrams of the first node comprises configuring the portion of graphical code to execute in response to the one or more user interface events to which the sub-diagram corresponds (objects can be contained within other objects and each object displayed on the GUI can have user interface events specified by the user, which in turn as has attached program code that executes in response to an event such as user selection) (column 10, line 48-column 11, line 3 and column 14, line 56-column 17, line 24).

Referring to claims 4 and 35, Cain teaches wherein for each portion of graphical code, displaying a portion of graphical code within one of the sub-diagrams of the first node comprises displaying the one or more nodes of the portion of graphical code within one of the sub-diagrams of the first node (objects can be contained within other objects and each object has attached program code that executes in response to an event such as user selection) (column 10, line 48-column 11, line 3 and column 14, line 56-column 17, line 24).

Referring to claims 5, 51 and 64, Cain teaches wherein the block diagram comprises a data flow block diagram (as shown by the “Object Tree” in Figure 5D for example).

Referring to claims 6, 25, 37, 49, 50 and 54, Cain teaches executing the graphical program (executing the program via the run mode) (column 11, lines 17-29); wherein one or more user interface events which the first node is configured to receive are generated during execution of the graphical program, wherein the generating the first user interface comprises generating the first user interface event in response to user input to the graphical user interface (users can change or attach new user interface events which the button will respond to via input on the graphical user interface, as described with relation to Figures 4D-4E) (column 11, line 30-

column 12, line 46); wherein the method further comprises executing one of the portions of graphical code associated with the first node in response to each of the one or more user interface events which the first node is configured to receive being generated during execution of the graphical program (during execution of the program, i.e. in the run mode, the button object will respond to user defined interface events such as generation of a push-button event upon clicking on the button; attaching program code to a button such that the code executes at run-time when the button is clicked) (column 10, line 48-column 11, line 3 and column 11, line 30-column 12, line 46).

Referring to claim 7, Cain teaches wherein the one or more user interface events generated during execution of the graphical program are generated in response to user input to the graphical user interface of the graphical program (such as user clicking a button) (column 10, line 48-column 11, line 3 and column 11, line 30-column 12, line 46).

Referring to claim 10, Cain teaches wherein the configuring the first node to receive the one or more user interface events comprises configuring the first node to receive notification when the one or more user interface events are generated during execution of the graphical program (the node, i.e. the button is notified of user event signals such as mouse button clicks and drags in order to execute the associated method in response to the user event) (column 3, lines 24-56).

Referring to claim 11, Cain teaches wherein the configuring the first node to receive the one or more user interface events comprises configuring the first node to receive information specifying occurrences of the one or more user interface events during execution of the graphical program (the node, i.e. the button is notified of when an user interface event such as mouse

button clicks and drags has occurred in order to execute the associated method in response to the user event) (column 3, lines 24-56).

Referring to claims 12, 40, 57 and 70, Cain teaches displaying the first portion of graphical source code within the first node in response to user input, wherein the first portion of graphical source code associated with the first node is visibly displayed within the first node of the block diagram of the graphical program (for example, Figures 4E and 4F) (column 10, line 48- column 13, line 17).

Referring to claims 13, 46, 48 and 62-63, Cain teaches wherein the configuring the first node to receive the one or more user interface events comprises configuring the first node to receive a first user interface event; wherein the first user interface event explicitly specifies a first user interface element of the graphical user interface and an action performed on the first user interface element (configuring the node, i.e. the button to display a “Hell. World!” message upon an event such as user selection) (column 10, line 65-column 11, line 29).

Referring to claims 14 and 47, Cain teaches wherein the first interface element comprises one of an indicator, a control, a menu element and a window (the button is programmed to display a dialog box window upon user clicking the button) (column 10, line 65-column 11, line 3).

Referring to claims 15, 44 and 60, Cain teaches displaying a first graphical user interface dialog for configuring the first node (Figures 4E and 4D); wherein the receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input to the first graphical user interface dialog to explicitly specify the

one or more user interface events (specifying user interface events allows users to define and attach method to buttons) (column 11, line 30-column 12, line 46).

Referring to claims 16 and 71, as best understood by the examiner, Cain teaches displaying a second node for dynamically registering user interface events in the block diagram in response to user input (a plurality of nodes or objects such as buttons that can be created by the user) (column 14, line 30-column 17, line 24 and Figure 5D); wherein the receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input explicitly specifying a first user interface event to dynamically register during execution of the graphical program (Figure 4E shows the receiving of user input specifying methods corresponding to user interface events to be executed during the run-mode) (column 10, line 50-column 12, line 46), wherein the method further comprises configuring the second node to dynamically register the first user interface event during execution of the graphical program (Figure 4E shows the receiving of user input specifying methods corresponding to user interface events to be executed during the run-mode, such as the first user interface event of clicking the button; furthermore, each node, or button can have a plurality of associated events such as clicking the button) (column 10, line 50-column 12, line 63 and Figures 6E and 6F); wherein after dynamically registering the first user interface event, the first node is operable to receive the first user interface event (program code corresponding to how a node, i.e. button respond to specified user interface events are attached to the button so that in the run mode, the button is operable to receive user interface events such as clicking the button and respond accordingly) (column 10, line 50-column 12, line 46).

Referring to claim 17, Cain teaches wherein the configuring the second node dynamically register the first user interface event during execution of the graphical program comprises connecting the second node to the first node in response to user input (user selection of objects and buttons creates the interconnected nodes shown in the “Object Tree” of Figure 5D) (column 16, line 31-column 17, line 67).

Referring to claim 18, Cain teaches wherein the one or more user interface events specified by the third user input includes a first user interface event (specifying events such as the events shown in Figure 4E) (column 3, lines 24-56 and column 10, line 50-column 12, line 49); wherein the method further comprises displaying a second node dynamically un-registering user interface events in the block diagram in response to user input; configuring the second node to dynamically un-register the first user interface event during execution of the graphical program (as shown in Figure 4E, events can be un-registered, i.e. deleted so that they are not part of the program code for the button during run time); wherein before the dynamically un-registering the first user interface event, the first node is operable to receive the first user interface event; wherein after the dynamically un-registering the first user interface event, the first node is not operable to receive the first user interface event (users can receive events listed in the built-in method box shown in Figure 4E before unregistering, i.e. deleting, an event, at which time the event will no longer be a part of, i.e. responded to by the button) (column 11, line 4-column 12, line 49).

Referring to claim 22, Cain teaches wherein the one or more programmatic events comprises one or more of a user interface event, a system event, a timer event, an event

generated in response to data acquired from a device (programming buttons to respond to user interface events such as user clicking a button) (column 10, line 48-column 11, line 3).

Referring to claims 24, 38-39, 52, 55-56 and 65, Cain teaches wherein the first portion of graphical code includes a plurality of nodes (a plurality of interconnected nodes, as shown in the “Object Tree” of Figure 5D), wherein the creating the first portion of graphical code in response to user input comprises arranging the plurality of nodes on a display and interconnecting the plurality of nodes in response to user input (user selection of objects and buttons creates the interconnected nodes shown in the “Object Tree” of Figure 5D) (column 16, line 31-column 17, line 67), wherein the plurality of interconnected nodes visually indicates functionality for responding to the one or more user interface events associated with the first portion of graphical code (the plurality of interconnected nodes visually indicate functionality such as containership relationships which controls how user interface events are handled) (column 14, line 56-column 17, line 67).

Referring to claims 41, 58, 78 and 83, Cain teaches displaying information explicitly identifying a plurality of user interface events, wherein the receiving the user input explicitly specifying the one or more user interface events to associate with the first portion of graphical code comprises receiving user input selecting the one or more user interface events from the displayed information (users can select and custom create user interface events from the displayed events shown in window 461 of Figure 4E) (column 11, line 55-column 12, line 13).

Referring to claims 42 and 59, Cain teaches associating a second portion of graphical source code with the first node in response to user input (each node, or button, can have a plurality of default and custom methods with associated program code that execute in response to

events) (column 11, line 55-column 12, line 63 and Figures 6E and 6F); associating a second user interface event with the first node in response to user input explicitly specifying the second user interface event and configuring the second portion of graphical source code associated with the first node to execute in response to the second user interface event associated with the first node (each node, or button can have a plurality of associated events such that a plurality of program codes corresponding to the plurality of user-defined methods are executed in response to the plurality of events) (column 11, line 55-column 12, line 63 and Figures 6E and 6F).

Referring to claims 43, 68 and 85, Cain teaches wherein the receiving the user input explicitly specifying the one or more user interface events to associate with the first portion of graphical code comprises receiving user input specifying names of the one or more user interface events to associate with the first portion of graphical code (users can create new user interface events, i.e. name new events to be associated with a button via window 461 of Figure 4E; users can also specify and change the names of the buttons) (column 11, line 4-column 12, line 46).

Referring to claims 67 and 84, Cain teaches displaying a list of user interface events, wherein the receiving the user input explicitly specifying the one or more user interface events to associate with the first portion of graphical code comprises receiving user input selecting the one or more user interface events from the displayed list of user interface events (users can select and custom create user interface events from the displayed list of events shown in window 461 of Figure 4E) (column 11, line 55-column 12, line 13).

Referring to claim 69, Cain teaches wherein the associating the one or more portions of graphical code with the first node comprises associating a first portion of graphical code and a second portion of graphical code with the first node (each node, or button can have a plurality of

associated events such that a plurality of program codes corresponding to the plurality of user-defined methods are executed in response to the plurality of events) (column 11, line 55-column 12, line 63 and Figures 6E and 6F); wherein the first node comprises a plurality of sub-diagrams (objects contained within other objects) (“Object Tree” shown in Figure 5D); wherein associating the first portion of graphical code with the first node comprises displaying the first portion of graphical code within a first sub-diagram of the first node; and wherein associating the second portion of graphical code with the first node comprises displaying the second portion of graphical code within a second sub-diagram of the first node (objects contained with other objects inherits the events, i.e. program code associated with events, from the parent object) (column 14, line 56-column 17, line 24).

Referring to claim 74, Cain teaches wherein each of the portions of graphical code is located separately from the first node (for example, as shown in Figure 5B, each of the objects in the object tree, and therefore, their associated portion of program code are an individual and separate entity).

Referring to claim 75, Cain teaches wherein for each of the portions of graphical code, the associating the portion of graphical code with the first node comprises associating the portion of graphical code with one or more of the user interface events which the first node is configured to receive, wherein the portion of graphical code comprises one or more nodes for responding to the one or more user interface events with which the portion of graphical code is associated (attaching program code specifying how the button will respond to a specific user interface event, to the button) (column 10, line 50-column 11, line 29).



Referring to claim 76, Cain teaches wherein the receiving third user input explicitly specifying the one or more user interface events to configure for the first node comprises receiving user input explicitly specifying a first user interface event and a second user interface event (each node, or button can have a plurality of associated events such that a plurality of program codes corresponding to the plurality of user-defined methods are executed in response to the plurality of events) (column 11, line 55-column 12, line 63 and Figures 6E and 6F); wherein the associating the one or more portions of graphical code with the first node comprises associating a first portion of graphical code with the first node, wherein the first portion of graphical code comprises a plurality of interconnected nodes visually indicating functionality for responding to the first user interface event; wherein the associating the one or more portions of graphical code with the first node further comprises associating a second portion of graphical code with the first node, wherein the second portion of graphical code comprises a plurality of interconnected nodes visually indicating functionality for responding to the second user interface event (each node, or button can have a plurality of associated events such that a plurality of program codes corresponding to the plurality of user-defined methods are executed in response to the plurality of events; furthermore, the plurality of interconnected nodes shown in Figure 5D visually indicate functionality such as containership relationships which controls how user interface events are handled) (column 11, line 55-column 12, line 63, column 14, line 56-column 17, line 67 and Figures 6E and 6F).

Referring to claim 77, Cain teaches executing the graphical program (column 11, lines 17-29); executing the first portion of graphical code associated with the first node in response to the first user interface event being generated during execution of the graphical program; and

executing the second portion of graphical code associated with the first node in response to the second user interface event being generated during execution of the graphical program (each node, or button can have a plurality of associated events such that a plurality of program codes corresponding to the plurality of user-defined methods are executed in response to the plurality of events) (column 11, line 55-column 12, line 63 and Figures 6E and 6F).

Referring to claim 79, Cain teaches wherein the receiving the third user input explicitly specifying the one or more user interface events to configure for the first node comprises graphically connecting a plurality of objects to the first node in response to user input, wherein each object directly represents a user interface event (plurality of objects are connected to other objects, as shown in Figure 5D, each object has an associated user interface events, such as built-in events) (column 11, lines 55-63 and column 14, line 30-column 17, line 24).

Referring to claim 82, Cain teaches creating the portions of graphical code in response to user input (creating code associated with method in response to users creating methods) (column 10, lines 50-58 and column 12, lines 7-29), wherein for each portion of graphical code, creating the portion of graphical code comprises including one or more nodes in the portion of graphical code in response to user input, wherein the one or more nodes are operable to respond to one or more of the user interface events which the first node is configured to receive (the nodes or buttons will respond to user interface events such as clicking the button according to the associated program code that executes with the button is clicked) (column 10, line 50-column 12, line 46).

Referring to claim 87, Cain teaches a method comprising creating a first portion of graphical code in response to user input (creating customized buttons with associated code for

responding to specific user interface events) (column 10, line 50-column 12, line 46 and Figure 4E), wherein the first portion of graphical code comprises one or more nodes that visually indicate functionality for responding to programmatic events generated during execution of the graphical program (the plurality of interconnected nodes visually indicate functionality such as containership relationships which controls how user interface events are handled) (column 14, line 56-column 17, line 67); and configuring the graphical program to dynamically register a first programmatic event during execution of the graphical program (Figure 4E shows the receiving of user input specifying methods corresponding to user interface events to be executed during the run-mode) (column 10, line 50-column 12, line 46), wherein dynamically registering the first programmatic event comprises dynamically associating the first programmatic event with the first portion of graphical code (program code corresponding to how a node, i.e. button respond to specified user interface events are attached to the button so that it executes when the button is selected in the run mode) (column 10, line 50-column 12, line 46); wherein the dynamically registering the first programmatic event causes the first portion of graphical code to execute in response to the first programmatic event being generated (attaching program code to the button so that when the first programmatic event of the button being clicked is received, the code executes) (column 10, line 50-column 11, line 3).

Referring to claim 88, Cain teaches receiving user input specifying the first programmatic event to be dynamically registered during execution of the graphical program (Figure 4E shows the receiving of user input specifying methods corresponding to user interface events to be executed during the run-mode) (column 10, line 50-column 12, line 46); wherein the graphical program is configured to dynamically register the first programmatic event during execution of

the graphical program in response to the user input specifying the first programmatic event (program code corresponding to how a node, i.e. button respond to specified user interface events are attached to the button so that it executes when the button is selected in the run mode) (column 10, line 50-column 12, line 46).

Referring to claim 89, Cain teaches wherein the configuring the graphical program to dynamically register the first programmatic event during execution of the graphical program comprises configuring a node in the graphical program to dynamically register the first programmatic event during execution of the graphical program (program code corresponding to how a node, i.e. button respond to specified user interface events are attached to the button so that it executes when the button is selected in the run mode) (column 10, line 50-column 12, line 46).

### *Claim Rejections - 35 USC § 103*

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable Cain et al. U.S. Patent 5,651,108 (hereinafter "Cain"), as applied to claims 1 and 6 above, and Zizzo U.S. Patent 6,578,174.

Referring to claims 8 and 9, Cain teaches all of the limitations as applied to the claims 1 and 6 above. However, Cain fails to explicitly teach a block diagram executing on a first reconfigurable instrument and the graphical user interface displayed on a display of a second, connected computer system. Zizzo teaches a method for providing a graphical user interface for creating a graphical program (providing tools for designing circuits) (Zizzo: column 18, lines 36-38) similar to that of Cain. In addition, Zizzo further teaches the design executes on a first reconfigurable instrument (server computer system) and the graphical user interface is displayed on a display of a second computer system (the circuit design executes on a central server, or first computer system while a plurality of user, or second computer systems, connected to the server through a network displays the user interface used in designing the circuit diagram) (Zizzo: column 4, lines 50-60). It would have been obvious to one of ordinary skill in the art, having the teachings of Cain and Zizzo before him at the time the invention was made, to modify system for executing and displaying block diagrams of Cain to include the use of a client/server network system in executing and designing diagrams, taught by Zizzo. One would have been motivated to make such a combination in order to allow design tools to be readily available and easily used on a variety of computing platforms and operating systems.

### ***Response to Arguments***

8. Applicant's arguments with respect to claims 1-25, 32-44, 46-60 and 62-89 have been considered but are moot in view of the new ground(s) of rejection.

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

### *Conclusion*

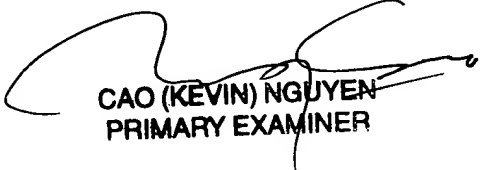
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ting Zhou whose telephone number is (571) 272-4058. The examiner can normally be reached on Monday - Friday 7:00 am - 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached at (571) 272-4048. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Art Unit: 2173

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TZ



CAO (KEVIN) NGUYEN  
PRIMARY EXAMINER